38481-8043.US01 *Patent*

UNITED STATES PATENT APPLICATION
FOR

COMMON COMMON OBJECT

INVENTORS:
MARIA THERESA BARNES-LEON
NARDO B. CATAHAN JR.
RICHARD EXLEY

PREPARED BY:

PERKINS COIE, LLP
101 JEFFERSON DRIVE
MENLO PARK, CALIFORNIA
(650) 838-4300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EV 336042898 US

Date of Deposit March 24, 2004

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA. 22313-1450.

Carina Tan
(Typed or printed name of person mailing paper or fee)

Carina M Tan
(Signature of person mailing paper or fee)

COMMON COMMON OBJECT

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 60/457,493 filed March 24, 2003, entitled, "COMMON COMMON OBJECT," by Barnes-Leon et al., and which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The present invention is directed to the field of data modeling, and more specifically to aspects of reusable data types that can be referenced by other data objects.

BACKGROUND

[0003] An enterprise may employ various systems to manage various aspects of human resources and enterprise resources. The various systems can include Human Resource Management (HRM) systems, Employee Relationship Management (ERM) systems, Enterprise Resources Planning (ERP) systems, supply chain management (SCM) and warehouse management (WMS), and custom applications for the purpose of sharing data. Such an enterprise system is herein referred to as a multi-application integration system (MAIS). The various systems in the MAIS need to communicate data to each other. However, the users of enterprise data in the back-office typically store data in forms usable by the back-office computerized system, which often differ significantly from the forms usable with front-office computerized systems.

[0004] Thus, when some or all aspects of enterprise data are managed by both back-office and front-office computerized systems, there is a need to synchronize the enterprise data in both computerized systems.

[0005] Thus, in order for front-office computerized systems to communicate with back-office computerized systems that are already being used, the user must manually regenerate data from the back-office computerized systems in forms usable by the front-office computerized systems. Such manual regeneration has several significant disadvantages, including: (1) it is often expensive; (2) it often requires a substantial amount of time to complete; (3) it must be repeated each time data changes in either the back-office system or the front-office system; and (4) it is prone to errors.

[0006] In view of the foregoing, an automated and efficient approach for transforming data used by a back-office computerized system for use by a front-office computerized system, or vice versa, is needed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1A is a high-level network diagram showing aspects of a computerized environment in which the facility operates, according to certain embodiments.

[0008] FIG. 1B is a block diagram showing some of the components typically incorporated in at least some of the computer systems and other devices on which the facility executes.

[0009] FIG. 2 shows the application instance structure 200.

[0010] FIG. 3 shows the fault handler input structure 300.

[0011] FIG. 4 shows the fault handler output structure 400.

[0012] FIG. 5 shows the fault transformer input structure 500

[0013] FIG. 6 shows the fault transformer output structure 600.

[0014] FIG. 7 shows the list of application instance structure 700.

[0015] FIG. 8 shows the list of application type structure 800.

[0016] FIG. 9 shows the list of ID Cross-reference structure 900.

[0017] FIG. 10 shows the list of ID Cross-reference data structure 1000.

[0018] FIG. 11 shows the list of message definition structure 1100.

[0019] FIG. 12 shows the list of message text structure 1200.

[0020] FIG. 13 shows the list of value cross-reference structure 1300.

[0021] FIG. 14 shows the value cross-reference data structure 1400.

[0022] FIG. 15 shows the message text element 1500.

[0023] FIG. 16 shows the message set structure 1600.

[0024] FIG. 17 shows the activity type structure 1700.

[0025] FIG. 18 shows the address type structure 1800.

[0026] FIG. 19 shows the communication data type structure 1900.

[0027] FIG. 20 shows the payment card type structure 2000.

[0028] FIG. 21 shows the alternate identification type structure 2100.

[0029] FIG. 22 shows the data cleansing data type structure 2200.

DETAILED DESCRIPTION

[0030] All changes in the enterprise information need to be captured and made accessible to all relevant computer applications that the enterprise uses to manage various aspects of enterprise resources. Thus, a common data storage model is needed for enabling users of the relevant computer applications to have the same view of the enterprise information across the various computer applications.

[0031] According to certain embodiments, the common data storage model utilizes common objects that provide defined data structures that can be used as conduits for passing enterprise information from one computerized system to another in the enterprise multi-application integration system (MAIS). Such a data structure is a common structure that can be mapped to multiple distinct enterprise systems purchased from different vendors. Such a common data storage model is herein referred to as a common object data model or an MAIS data model.

[0032] One aspect of the common object data model is the design and utilization of "Common" common objects. In other words, the "Common" common objects provide data types that can be shared by a multiplicity of common objects that are in the common object data model.

[0033] The data types associated with Common common objects are defined in a Common common object schema, herein referred to as "common.xsd". All integration application processes in MAIS have the common.xsd available for use within such integration application processes.

The common.xsd provides reusable data types that can be referenced by any common object within the common object data model. Instead of defining redundant data types that are local to each common object, the common.xsd provides global data types that can be used by all common objects, thus reducing the complexity of the data model, and allowing for simpler changes of data types in the future (if needed). The alternative is to define all elements within each common object, but to do so would increase maintenance costs and decrease standardization.

The common.xsd is not meant to be the data transport schema by itself. It is a library (repository) of commonly used data types. The data types within common.xsd provide standard definitions of data elements that can be reused by other common objects.

Some of the data types defined in the common.xsd are:

- Address data types
- Communication data types (phone numbers, email, etc.)
- Cross-reference data types (value cross reference, ID cross reference, etc.)

- Message data types (error messages, informational messages)
- Credit Card or payment card data types
- Alternate ID data types
- Fault Handling data types
- Activity data types
- Data Cleansing data types
- Application data types

For example, common objects in the common object data model that require address information simply refer to the address data type defined in common.xsd, rather than recreating an entire address format within each of the common objects that need address information. Thus, there may be many common objects referring to the same address data type defined in common.xsd. If it is later determined that the addition of more elements within the address data type would be beneficial, the data type would be modified in common.xsd, and all of the common objects that refer to the address data type will inherit the new definition of an address. On the other hand, if the address format were defined in every common object that needs address information, then any changes to the address format would require that all such common objects be updated manually. The common.xsd, as used within MAIS, significantly reduces maintenance efforts and increases standardization of the data model.

Thus, the design of the Common common object is suitably adapted to evolve for enhancing the utility of the common object data model. Each reusable data type has a clearly defined native data type (e.g. string, date, integer, float, double, etc.) that corresponds to XML standards.

[0034] When enterprise information is passed from the back-office enterprise system to the front-office enterprise system, then the back-office enterprise system is referred to as the source system and the front-office enterprise system is referred to as the target system. On the other hand, when enterprise information is passed from the front-office enterprise system to the back-office enterprise system, then the front-office enterprise system is referred to as the source system and the back-office enterprise system is referred to as the target system.

[0035] A software facility (hereafter "the facility") for automatically converting enterprise information, is described. In some embodiments, the facility converts enterprise information from a form used by the source system to a form used by the target system.

[0036] In some embodiments, such as embodiments adapted for converting enterprise information in the first source format, the facility converts enterprise information by converting the enterprise information that is in the first source format into an intermediate format. The intermediate format includes a plurality of common data type elements that are adapted to be shared across a plurality of data objects in the intermediate format. The intermediate format is then used to convert the enterprise information into the target format.

[0037] By performing such conversions, embodiments of the facility enable a user of a first computerized system who has stored enterprise information in a first format for use by the first computerized system to readily make the stored enterprise information available for use in a second computerized system that utilizes a second format in a cost-efficient and time-efficient manner.

[0038] FIG. 1A is a high-level network diagram showing aspects of a typical hardware environment in which the facility operates. FIG. 1A shows a source system 110, a target system 130, an integration server 120 and a network 150. Source system 110 stores enterprise information in a source format. There may be more than one source system. Target system 130 stores enterprise information in a target format. There may be more than one target system.

[0039] The facility (not shown) converts some or all the enterprise information that is in the source format into the target format by using an intermediate format of the enterprise information. In certain embodiments, such conversions are performed with the aid of one or more other computer systems, such as integration server system 120. Components of the facility may reside on and/or execute on any combination of these computer systems, and intermediate results from the conversion may similarly reside on any combination of these computer systems.

[0040] The computer systems shown in FIG. 1A are connected via network 150, which may use a variety of different networking technologies, including wired, guided or line-of-sight optical, and radio frequency networking. In some embodiments, the network includes the public switched telephone network. Network connections established via the network may be fully-persistent, session-based, or intermittent, such as packet-based. While the facility typically operates in an environment such as is shown in FIG. 1A and described above, those skilled in the art will appreciate the facility may also operate in a wide variety of other environments.

[0041] FIG. 1B is a block diagram showing some of the components typically incorporated in at least some of the computer systems and other devices on which the facility executes, including some or all of the server and client computer systems shown in FIG. 1A. These computer systems and devices 100 may include one or more central processing units ("CPUs") 101 for executing computer programs; a computer memory 102 for storing programs and data -- including data structures -- while they are being used; a persistent storage device 103, such as a hard drive, for persistently storing programs and data; a computer-readable media drive 104, such as a CD-ROM drive, for reading programs and data stored on a computer-readable medium; and a network connection 105 for connecting the computer system to other computer systems, such as via the Internet, to exchange programs and/or data -- including data structures. While computer systems configured as described above are typically used to support the operation of the facility, those skilled in the art will appreciate that the facility may be implemented using devices of various types and configurations, and having various components.

[0042] It will be understood by those skilled in the art that the facility may transform enterprise information from a number of different source systems and from a number of different source software packages to a number of target systems and/or to a number of target software packages.

[0043] The intermediate data structures used by the facility include common common data structures. Common common data structures are reusable data types that can be referenced by other intermediate data structures. Common common data structures include one or more elements comprising: an application element, a fault handler input element, a fault handler output element, a fault transformer input element, a fault transformer output element, a list of application instance element, a list of application type element, a list of ID cross-reference element, a list of ID cross-reference data element, a list of message definition element, a list of message text element, a list of value cross-reference element, and a list of value cross-reference data element, a message element, a message set element, an activity type element, an address type element, an alternate ID type element, a communication data type element, a data cleansing data type element, and a payment card type element.

[0044] FIG. 2 shows that the application instance structure 200 may include one or more of: an application instance name element 202, an application type name element 204, and an application instance description element 206.

[0045] FIG. 3 shows that the fault handler input structure 300 may include one or more of: an error type element 302, an error language element 304, an error severity element 306, an error flow name element 308, an error flow context element 310, a process name element 312, a message set element 314, a plurality of message text sub-elements 316, and a plurality of child message set sub-elements 318. Each child message set sub-element can include other message set sub-elements 320.

[0046] FIG. 4 shows that the fault handler output structure 400 may include a message text element 402.

[0047] FIG. 5 shows that the fault transformer input structure 500 may include one or more of: an error type element 502, an error language element 504, an error severity element 506, an error flow name element 508, an error flow context element 510, a process name element 512, a message set element 514, a plurality of message text sub-elements 516, and a plurality of child message set sub-elements 518. Each child message set sub-element can include other message set sub-elements 520.

[0048] FIG. 6 shows that the fault transformer output structure 600 may include a message text element 602.

[0049] FIG. 7 shows that the list of application instance structure 700 may include one or more of: a plurality of application instance definition elements 702, an application instance name sub-element 704, an application type name sub-element 706, an application instance description sub-element 708, a list of one-to-many ID cross-reference sub-element 710, a plurality of ID cross-reference sub-elements 712, an ID cross-reference name sub-element 714, and an ID cross-reference description sub-element 716.

[0050] FIG. 8 shows that the list of application type structure 800 may include one or more of: a plurality of application type elements 802, an application type name sub-element 804, and an application type description sub-element 806.

[0051] FIG. 9 shows that the list of ID Cross-reference structure 900 may include one or more of: a plurality of ID cross-reference elements 902, an ID cross-reference name sub-element 904, and an ID cross-reference description sub-element 906.

[0052] FIG. 10 shows that the list of ID Cross-reference data structure 1000 may include one or more of: a plurality of ID cross-reference elements 1002, a plurality of application instance sub-elements 1004, and a plurality of application ID sub-elements 1006.

[0053] FIG. 11 shows that the list of message definition structure 1100 may include one or more of: a plurality of message definition elements 1102; a message code sub-element 1104, a message description sub-element 1106, a message corrective action sub-element 1108, and a message argument name sub-element 1110.

[0054] FIG. 12 shows that the list of message text structure 1200 may include a plurality of message text elements 1202.

[0055] FIG. 13 shows that the list of value cross-reference structure 1300 may include one or more of: a plurality of value cross-reference elements 1302, a value cross-reference name sub-element 1304, and a value cross-reference description sub-element 1306.

[0056] FIG. 14 shows that the value cross-reference data structure 1400 may include one or more of: a plurality of value cross-reference elements 1402, a plurality of application type sub-elements 1404, and a plurality application value sub-elements 1406.

[0057] FIG. 15 shows the message text element 1500.

[0058] FIG. 16 shows that the message set structure 1600 may include one or more of: a plurality of message text elements 1602, and a plurality of child message set elements 1604. Each child message set element can include a message set sub-element 1606.

[0059] FIG. 17 shows that the activity type structure 1700 may include one or more of: an activity published code element 1702, an activity comment element 1704, an activity duration element 1706, an activity end date element 1712, an activity number element 1722, an activity reason code element 1724, an activity start date element 1726, an activity task description element 1736, an activity type code element 1738, an activity planned duration sub-element 1708, an activity actual duration sub-element 1710. End date element 1712 includes an activity actual date sub-element 1714, an activity actual time sub-element 1716, an activity planned date sub-element 1718, and an activity planned time sub-element 1720. Start date element 1726 includes an activity actual date sub-element 1728, an activity actual time sub-element 1730, an activity planned date sub-element 1732, and an activity planned time sub-element 1734.

[0060] FIG. 18 shows that the address type structure 1800 may include one or more of: a plurality of address line elements 1802, 1804, 1806, 1808, an address city element 1810, an address state code element 1812, an address county element 1814, an address province element 1816, an address country code element 1818, an address house number element 1820, an address house prefix element 1822, an address house suffix element 1824, an address postal code element 1826, an address street direction element 1828, an address street name element 1830, an address street number element 1832, an address street prefix number element 1834, an address street prefix type element 1836, an address street suffix element 1838, an address thoroughfare element 1840, an address list of location designator element 1842, a plurality of address location designator sub-elements 1844, an address location designator name sub-element 1846, and an address location designator value sub-element 1848.

[0061] FIG. 19 shows that the communication data type structure 1900 may include one or more of: a list of phone number element 1902, a list of email element 1920, a list of web page element 1928, a custom communication data element 1936, a phone number sub-element 1904, an email sub-element 1922, a web page sub-element 1930, a phone number area code sub-element 1906, a phone number country code sub-element 1908, a phone number full number sub-element 1910, a phone number extension sub-element 1912, a phone number international access code sub-element 1914, a phone number number sub-element 1916, a phone number type code sub-element 1918, an email type sub-element 1924, an email address sub-element 1926, a web page type sub-element 1932, and a web page address sub-element 1934.

[0062] FIG. 20 shows that the payment card type structure 2000 may include one or more of: a payment card type element 2002, a payment card number element 2004, a payment card holder element 2006, a payment card expiration year element 2008, a payment card expiration month element 2010, and a payment card verification number element 2012.

[0063] FIG. 21 shows that the alternate identification type structure 2100 may include one or more of: an ID element 2102, and an ID type element 2104.

[0064] FIG. 22 shows that the data cleansing data type structure 2200 may include a disable cleansing flag element 2202.

[0065] It will be appreciated by those skilled in the art that the above-described facility may be straightforwardly adapted or extended in various ways. For example, the facility may be used to transform various other kinds of enterprise information, and may be used to transform enterprise information between a variety of other formats.

[0066] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what the invention is and what is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any express definitions set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.